AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Jae-Yoel KIM, et al.

Examiner: Colin, Carl G.

Serial No.: 09/611,518

Group Art Unit: 2136

Filed: July 7, 2000

Docket: 678-509 (P9463)

For: **APPARATUS AND METHOD FOR GENERATING SCRAMBLING CODE IN UMTS MOBILE COMMUNICATION SYSTEM**

Dated: September 10, 2007

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

TRANSMITTAL OF APPELLANTS' BRIEF ON APPEAL

Sir:

Enclosed please find APPELLANTS' BRIEF.

Also enclosed is a check in the amount of $500.00 to cover the appeal fee.

If the enclosed check is insufficient for any reason or becomes detached, please charge the required fee under 37 C.F.R. §1.17 to Deposit Account No. 50-4053. Also, in the event any additional extensions of time are required, please treat this paper as a petition to extend the time as required and charge Deposit Account No. 50-4053. TWO COPIES OF THIS SHEET ARE ENCLOSED.

Respectfully submitted,

Paul J. Farrell
Reg. No.: 33,494
Attorney for Applicant(s)

THE FARRELL LAW FIRM
333 Earle Ovington Blvd., Suite 701
Uniondale, New York 11553
516-228-3565

CERTIFICATE OF MAILING 37 C.F.R. §1.8(a)

I hereby certify that this correspondence (and any document referred to as being attached or enclosed) is being deposited with the United States Postal Service as first class mail, postage paid in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450, Mail Stop Appeal Brief-Patents on September 10, 2007.

Dated: September 10, 2007

Michael J. Musella

Attorney Docket No.: 678-509 (P9463)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT(S): Jae-Yoel KIM, et al.

GROUP ART UNIT: 2136

APPLICATION NO.: 09/611,518

EXAMINER: Colin, Carl G.

FILING DATE: July 7, 2000

DATED: September 10, 2007

FOR: APPARATUS AND METHOD FOR GENERATING SCRAMBLING CODE IN UMTS MOBILE COMMUNICATION SYSTEM

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' BRIEF ON APPEAL

REAL PARTY IN INTEREST

The real party in interest is Samsung Electronics Co, Ltd, the assignee of the subject application, having an office at 416, Maetan-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do, Republic of Korea.

RELATED APPEALS AND INTERFERENCES

To the best of Appellants' knowledge and belief, there are no currently pending related appeals, interferences or judicial proceedings.

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8 (a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Commissioner for Patents, Alexandria, VA 22313-1450, Mail Stop Appeal Brief-Patents on September 10, 2007.

Dated: September 10, 2007

Michael J. Musella

STATUS OF CLAIMS

The original application filed on July 7, 2000 contained Claims 1-30. In a Response filed June 28, 2004, Claim 26 was amended. In a Response filed January 10, 2005, Claims 1 and 21 were amended, Claims 2-20 and 22-30 were cancelled, and Claims 31-53 were newly added. In a Response filed September 30, 2005, Claims 1, 21, 40-46, 48 and 51 were amended. In a Response filed April 28, 2006, Claims 48-53 were cancelled, and Claims 54-70 were newly added. In a Response filed December 11, 2006, Claim 54 was amended.

Thus, Claims 1, 21, 31-47 and 54-70 are pending in the Appeal. Claims 1, 21, 54, 59 and 65 are in independent form.

For the purposes of this appeal, Claims 1 and 31-37 stand or fall together; Claims 21 and 38-47 stand or fall together; Claims 54-58 stand or fall together; Claims 59-64 stand or fall together; and, Claims 65-70 stand or fall together.

STATUS OF AMENDMENTS

To date, all of the amendments to the claims have been entered. Thus, the Appendix to this Appeal Brief includes Claims 1, 21, 31-47 and 54-70, of which the status of Claims 1, 21, 31-47 and 54-70 are indicated as "Previously Presented", and the status of Claims 2-20, 22-30 and 48-53 are indicated as "Cancelled".

SUMMARY OF CLAIMED SUBJECT MATTER

The invention as recited in Claim 1 relates to a method for generating a primary scrambling code. The method includes generating a first m-sequence from a first m-sequence generator including first shift registers having first shift register values a_i, wherein i = 0 to c-1 and where c is

the total number of the registers. (Specification at page 19, line 18 to page 20, line 23, FIG. 10).[1] The method further includes generating a second m-sequence from a second m-sequence generator including second shift registers having values b_j, wherein j = 0 to c-1, and where c is the total number of the registers. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The method still further includes masking the first shift register values a_i with a first set of mask values K_i, wherein i = 0 to c-1 to generate a third m-sequence. (Specification at page 20, lines 14-23, FIG. 10). The method yet further includes adding the first m-sequence with the second m-sequence to generate a primary scrambling code. (Specification at page 20, lines 24-26, FIG. 10). The method still yet further includes adding the third m-sequence and the second m-sequence to generate a secondary scrambling code. (Specification at page 20, line 26 to page 21, line 19, FIG. 10). Finally, in the method the masking step shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code. (Specification at page 20, lines 9-23 and page 10, line 28 to page 11, line 15, Equation (1)).

The invention as recited in Claim 21 relates to a scrambling code generator. The scrambling code generator includes a first m-sequence generator to generate a first m-sequence by using a plurality of first registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first registers. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The scrambling code generator further includes a second m-sequence generator to generate a second m-sequence by using a plurality of second registers with second shift register values b_j, wherein j = 0 to c-1 and where c is the total number of second registers. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The scrambling code generator still further includes a masking section to mask the first shift register values a_i with a first set of mask values K_i to generate a third m-sequence, wherein i = 0 to c-1 to generate a third m-sequence. (Specification at page 20, lines 14-23, FIG. 10). The scrambling code generator yet further includes a first adder to add the first m-sequence and the second m-sequence to generate a primary scrambling code. (Specification at page 20, lines 24-26, FIG. 10). The scrambling code generator still yet further includes a second adder to add the third m-

[1] Although a citation for each feature of the claims is provided herein, Appellants note that support may be found elsewhere in the written description.

sequence and the second m-sequence to generate a secondary scrambling code. (Specification at page 20, line 26 to page 21, line 19, FIG. 10). Finally, in the scrambling code generator the masking section shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code. (Specification at page 20, lines 9-23 and page 10, line 28 to page 11, line 15, Equation (1)).

The invention as recited in Claim 54 relates to a method for generating scrambling codes in mobile communication system having a scrambling code generator. The method includes generating a $((K-1)*M+K)^{th}$ gold code as a K^{th} primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code. (Specification at page 18, line 25 to page 19, line 18, FIG. 9). The method further includes generating $((K-1)*M+K+1)^{th}$ through $(K*M+K)^{th}$ gold codes as secondary scrambling codes associated with the K^{th} primary scrambling code. (Specification at page 18, line 25 to page 19, line 18, FIG. 9). Finally, in the method an L^{th} Gold code is generated by adding an (L-1)-times shifted first m-sequence and a second m-sequence. (Specification at page 11, lines 6-15).

The invention as recited in Claim 59 relates to an apparatus for generating scrambling codes in mobile communication system having a scrambling code generator. The apparatus includes a first m-sequence generator to generate a first m-sequence. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The apparatus further includes a second m-sequence generator to generate a second m-sequence. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The apparatus still further includes at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence. (Specification at page 18, line 25 to page 19, line 18, and page 11, lines 6-15). Finally, in the apparatus K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code. (Specification at page 18, line 25 to page 19, line 18).

The invention as recited in Claim 65 relates to a method for generating scrambling codes in mobile communication system having a scrambling code generator. The method includes generating

a first m-sequence. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The method further includes generating a second m-sequence. (Specification at page 19, line 18 to page 20, line 23, FIG. 10). The method still further includes generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence. (Specification at page 18, line 25 to page 19, line 18, and page 11, lines 6-15). Finally, in the method K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code. (Specification at page 18, line 25 to page 19, line 18).

GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

Whether Claims 1, 21, 31-47 and 54-70 are unpatentable based on a provisional rejection under the judicially created doctrine of obviousness-type double patenting as being unpatentable over Claims 1-32 of copending U.S. patent application serial number 11/003,558.[2]

Whether Claims 1, 21, 31-47 and 54-70 are unpatentable under 35 U.S.C. §103(a) over U.S. 6,339,646 (Dahlman et al.) in view of WO 99/12284 (Dahlman) and U.S. 6,141,374 (Burns).

ARGUMENT

The Examiner provisionally rejected Claims 1, 21, 31-47 and 54-70 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over Claims 1-32 of copending application 11/003,558. The Examiner rejected Claims 1, 21, 31-47 and 54-70 under 35 U.S.C. §103(a) as being unpatentable over Dahlman et al. in view of Dahlman and Burns.

1. The provisional rejection of Claims 1, 21, 31-47 and 54-70 under the judicially created doctrine of obviousness-type double patenting only raises a potential obviousness-type double patenting problem and thus is not ripe for resolution

Claims 1, 21, 31-47 and 54-70 were provisionally rejected under the judicially created

[2] U.S. patent application serial number 11/003,558 is a continuation (CON) application claiming the benefit of the present application, i.e. U.S. 11/003,558 is a later filed application.

doctrine of obviousness-type double patenting as being unpatentable over Claims 1-32 of copending U.S. Application Serial No. 11/003,558.[3] Since the rejection is a provisional rejection and since there remain outstanding art rejections under §103, Appellants respectfully reserve the right to address this issue if and when all of the remaining art rejections are overcome. See M.P.E.P. §804.I.B.[4]

2. Independent Claim 1 is patentable over Dahlman et al. in view of Dahlman and Burns

Independent Claim 1 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[5]

Claim 1 recites a method for generating a primary scrambling code. The method includes generating a first m-sequence from a first m-sequence generator including first shift registers having first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the registers. The method further includes generating a second m-sequence from a second m-sequence generator including second shift registers having values b_j, wherein j = 0 to c-1, and where c is the total number of the registers. The method still further includes masking the first shift register values a_i with a first set of mask values K_i, wherein i = 0 to c-1 to generate a third m-sequence. The method yet further includes adding the first m-sequence with the second m-sequence to generate a primary scrambling code. The method still yet further includes adding the third m-sequence and the second m-sequence to generate a secondary scrambling code. Finally, in the method the masking step shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code.

Dahlman et al. discloses slotted mode code usage in a cellular communication system.[6]

Dahlman discloses a method for assigning spreading codes.[7]

Burns discloses a method and apparatus for generating multiple matched-filter pseudo-

---

[3] See Office Action dated March 7, 2007 at pp 4-5.
[4] M.P.E.P. §804.I.B states that the merits of a provisional rejection can be addressed by both the applicant and the examiner without waiting for the first patent to issue.
[5] See Office Action dated March 7, 2007 at pp 6-8.
[6] See Dahlman et al. at title and abstract.
[7] See Dahlman at title and abstract.

random noise vectors in a code divisional multiple access demodulator.[8]

2A. Claim 1 teaches that the L^{th} secondary scrambling code associated with the primary scrambling code results from adding the second m-sequence and L-times shifted first m-sequence

Claim 1 relates to a method of generating scrambling codes where an L^{th} secondary scrambling code associated with a primary scrambling code results from adding a second m-sequence and an L-times shifted first m-sequence. A clear understanding of Claim 1 can be characterized in the following statements: 1) In the second adding step a secondary scrambling code is the result of adding the second m-sequence and the third m-sequence; 2) In the masking step the third m-sequence is the result of masking the first shift register. The first shift register is the shift register used to generate that first m-sequence. The masking step produces the third m-sequence; 3) In the wherein clause the L^{th} secondary scrambling code is eventually generated when the masking step cyclically shifts the first m-sequence by L chips. The masking step produces the third m-sequence, the third m-sequence is the first m-sequence cyclically shifted by L chips, i.e. the L^{th} third m-sequence; and 4) Therefore, an L^{th} secondary scrambling code is generated by adding the second m-sequence to the L^{th} third m-sequence, the L^{th} third m-sequence being generated by cyclically shifting the first m-sequence by L chips during the masking step.

Claim 1 relates to generating related scrambling codes. The scrambling codes are generated from three (3) m-sequences: a first m-sequence, a second m-sequence, and a third m-sequence. The third m-sequence is generated using the first m-sequence, and are therefore related to each other. Two specific scrambling codes are generated from the 3 m-sequences: a primary scrambling code and a secondary scrambling code. Adding the first and second m-sequences generates the primary scrambling code. Adding the second and third m-sequences generates the secondary scrambling code. Since the third m-sequence is a cyclic shift of the first m-sequence, there is an implicit relation between the primary scrambling code and the secondary scrambling code.

The masking step processes as follows: the first m-sequence is masked to generate the third m-sequence; the first m-sequence is cyclically shifted by the masking step to generate the third m-sequence; if the cyclical shift is 1, then a 1-times shifted first m-sequence is generated as the third m-

[8] See Burns at title and abstract.

sequence; if the cyclical shift is 2, then a 2-times shifted first m-sequence is generated as the third m-sequence; and, if the cyclical shift is L, then an L-times shifted first m-sequence is generated as the third m-sequence. "L-times shifted first m-sequence" is another way of saying a third m-sequence generated from shifting the first m-sequence L times.

The third m-sequence is used to generate the secondary scrambling code. Therefore, if the third m-sequence is the 1-times shifted first m-sequence, then a 1st secondary scrambling code is generated; if the third m-sequence is the 2-times shifted first m-sequence, then a 2nd secondary scrambling code is generated; and, if the third m-sequence is the L-times shifted first m-sequence, then an Lth secondary scrambling code is generated.

Therefore, the L^{th} secondary scrambling code results from adding the second m-sequence and L-times shifted first m-sequence.

It is well known that Gold codes have no specific order. Likewise, in order to use Gold codes having no priority as primary and secondary scrambling codes, the currently used primary scrambling code and the corresponding secondary scrambling codes need to be identified to a base station and a mobile station. The claims of the present application drastically simplify this process, since if a primary scrambling code and the corresponding secondary scrambling codes are generated from only two predetermined m-sequences, the corresponding secondary scrambling codes as well as the primary scrambling code to be used in the base station can be simply generated by notifying the base station and the mobile station of only the primary scrambling code.

2B. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose masking the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips, neither reference, nor any combination thereof, can be used to render obvious Claim 1

Independent Claim 1 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[9]

Claim 1 recites, in part, "masking the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L

[9] See Office Action dated March 7, 2007, at page 6.

chips." None of the cited references teaches or discloses a masking process to generate a third m-sequence by cyclically shifting a first m-sequence.

The Examiner alleges that Dahlman et al. and Burns teach these features.[10]

The Examiner cites Burns as allegedly disclosing generating code sequences and masking to produce secondary sequences. The Examiner states that it would have been obvious to add the secondary sequence, i.e. the third m-sequence, with a second m-sequence to produce a secondary scrambling code. While Burns does teach a single masking circuit 303, this masking circuit is used to generate matched filter vectors, not to produce a secondary sequence from a first m-sequence.[11]

Additionally, Burns utilizes masking to generate a plurality of PN codes having different offsets.[12] Burns performs masking to a generated local PN code, while Claim 1 uses masking to shift a first m-sequence among two different m-sequences (i.e., a first m-sequence and a second m-sequence) to eventually generate a secondary scrambling code, which is different than in Burns.

Burns discloses a concept of a masking process.[13] More specifically, Burns discloses a method for managing a scrambling code in IS-95 or CDMA 2000.[14] In Burns, all base stations use an identical PN code (or a scrambling code), but a unique shift offset for each base station is applied to the PN code.[15] Accordingly, Burns generates a local PN code identically used by all base stations, and then performs masking for applying a unique offset to the local PN code for each base station. However, in Claim 1, the masking is applied to an m-sequence, not to the generated PN code (scrambling code), which is distinguishable from Burns.

Claim 1 recites that the first m-sequence is masked to produce a third m-sequence. Neither Burns nor Dahlman et al. teach or disclose masking a first m-sequence to generate a third m-sequence. Dahlman does not cure this defect.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses masking the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips.

---

[10] See Office Action dated March 7, 2007 at pages 6-8.
[11] See Burns at col. 8, lines 2-14.
[12] See Burns at col. 8, lines 2-14.
[13] See Burns at col. 3, lines 4-20.
[14] See Burns at col. 1, lines 10-15.
[15] See Burns at col. 8, lines 45-61, FIG. 4.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 1 of the present application, of masking the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips, Claim 1 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 1 under §103(a) must be reversed.

2C. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose adding the third m-sequence and the second m-sequence to generate a secondary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 1

Independent Claim 1 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[16]

Claim 1 recites, in part, "adding the third m-sequence and the second m-sequence to generate a secondary scrambling code." Claim 1 plainly recites the adding of two m-sequences to generate a scrambling code.

The Examiner alleges that Burns teaches these features.[17] Specifically, the Examiner alleges:[18]

> ...and further discloses adding the masking sequence with another sequence (column 3, line 40 through column 4, line5)...Burns suggests shifts are multiplied to produce new values then the combined masks are then added to produce new state values (column 8, lines 45-61).

In fact, Burns does not "suggest" that which the Examiner believes. Burns actually states at col. 3, line 40 – col. 4, line 5:

> The IS-95 system may augment the PN code sequence by inserting an extra value in the PN code sequence so that the PN code sequence is a multiple of 2. Additional logic (not shown in FIG. 1) inserts the extra value into each sequence following 14 consecutive 1's or 0's. The extra value renders a 2^{15} chip period PN

[16] See Office Action dated March 7, 2007 at page 6.
[17] See Office Action dated March 7, 2007 at pages 7-8.
[18] See Office Action dated March 7, 2007 at the bottom of page 7 and the middle of page 8.

sequence. Also, as is known in the art, a periodic bit sequence with a first code phase may be combined with another sequence to form the same periodic bit sequence with a second code phase. This process is known as masking. Consequently, a delayed, or offset, version of the sequence may be generated by modulo-2 addition of appropriate state bits of the shift register 102 with a selected mask. Additional logic for correcting the masked sequences may also be required if the PN code sequence is augmented.

And, Burns at col. 8, lines 45-61, states:

> Shift register 402 is loaded with a reference state as described with respect to FIG. 2. Then, for each clock cycle, the values $S=S_{[n:1]}$ of the shift register stages are multiplied by polynomial coefficients $g_{[n:1]}$ via gain amplifiers 404 and combined in modulo-2 adder 410 to provide new value s_0. This is a cyclic process. The value s_0 in modulo-2 adder 410 is then applied to the first element of the shift register 402 and the last element s_n is discarded. For each state of the shift register 402, a new state may be provided which corresponds to a value of the PN sequence shifted by an offset delay. Combining a state of shift register 102 with a corresponding mask value stored in mask register 412 generates this new state. The mask values $M=m_{[n-1:0]}$ are combined with the state of shift register 402 by combiners 414. The combined mask and register stage values are then modulo-2 added by adder 416 to provide the new state value o_i of the offset sequence $O_{[n:1]}$.

This first section of Burns is describing the apparatus illustrated in FIG. 1, and this second section of Burns is describing the apparatus illustrated in FIG. 4. Thus, the outputs of the shift registers of Burns are modulo-2 added to produce a single value. The apparatus of Burns does not add two (2) m-sequences. The single value of Burns is not and cannot be equated with adding the third m-sequence and the second m-sequence to generate a secondary scrambling code as recited in Claim 1.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses adding the third m-sequence and the second m-sequence to generate a secondary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 1 of the present application, of adding the third m-sequence and the second m-sequence to generate a secondary scrambling code, Claim 1 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 1 under §103(a) must be reversed.

2D. Independent Claim 1 is not rendered obvious by Dahlman et al. in view of Dahlman and Burns

The Examiner has failed to show that each and every element of Claim 1, and in as complete detail as is contained therein, are taught in or suggested by the prior art. The Examiner has failed to make out a prima facia case for an obviousness rejection, and thus Claim 1 is allowable.

3. Dependent Claims 31-37 are also patentable

Dependent Claim 31-37 were said to be unpatentable. Without conceding the patentability per se of dependent Claims 31-37, these claims are likewise believed to be allowable at least by virtue of their dependence on Claim 1.

4. Independent Claim 21 is patentable over Dahlman et al. in view of Dahlman and Burns

Independent Claim 21 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[19]

Claim 21 recites a scrambling code generator. The scrambling code generator includes a first m-sequence generator to generate a first m-sequence by using a plurality of first registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first registers. The scrambling code generator further includes a second m-sequence generator to generate a second m-sequence by using a plurality of second registers with second shift register values b_j, wherein j = 0 to c-1 and where c is the total number of second registers. The scrambling code generator still further includes a masking section to mask the first shift register values a_i with a first set of mask values K_i to generate a third m-sequence, wherein i = 0 to c-1 to generate a third m-sequence. The scrambling code generator yet further includes a first adder to add the first m-sequence and the second m-sequence to generate a primary scrambling code. The scrambling code generator still yet further includes a second adder to add the third m-sequence and the second m-sequence to generate a secondary scrambling code. Finally, in the scrambling code generator the masking section shifts the

[19] See Office Action dated March 7, 2007 at page 6, and pages 8-10.

first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code.

Dahlman et al. discloses slotted mode code usage in a cellular communication system.[20]

Dahlman discloses a method for assigning spreading codes.[21]

Burns discloses a method and apparatus for generating multiple matched-filter pseudo-random noise vectors in a code divisional multiple access demodulator.[22]

4A. Claim 21 teaches that the L^{th} secondary scrambling code associated with the primary scrambling code results from adding the second m-sequence and L-times shifted first m-sequence

Claim 21 relates to a scrambling code generator where an L^{th} secondary scrambling code associated with a primary scrambling code results from adding a second m-sequence and an L-times shifted first m-sequence.

As stated above in section 2A, neither Dahlman et al., Dahlman or Burns teach or disclose a scrambling code generator where an L^{th} secondary scrambling code associated with a primary scrambling code results from adding a second m-sequence and an L-times shifted first m-sequence.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses a scrambling code generator where an L^{th} secondary scrambling code associated with a primary scrambling code results from adding a second m-sequence and an L-times shifted first m-sequence.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 21 of the present application, of a scrambling code generator where an L^{th} secondary scrambling code associated with a primary scrambling code results from adding a second m-sequence and an L-times shifted first m-sequence, Claim 21 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 21 under §103(a) must be reversed.

---

[20] See Dahlman et al. at title and abstract.
[21] See Dahlman at title and abstract.

4B. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose a masking section to mask the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips, neither reference, nor any combination thereof, can be used to render obvious Claim 21

Claim 21 recites to a scrambling code generator having a masking section to mask the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips.

As stated above in section 2B, neither Dahlman et al., Dahlman or Burns teach or disclose a scrambling code generator having a masking section to mask the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses a scrambling code generator having a masking section to mask the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 21 of the present application, of a scrambling code generator having a masking section to mask the first shift register values with a first set of mask values to generate a third m-sequence, wherein the masking shifts the first m-sequence cyclically by L chips, Claim 21 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 21 under §103(a) must be reversed.

4C. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose an adder for adding the third m-sequence and the second m-sequence to generate a secondary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 21

Claim 21 recites to a scrambling code generator having an adder for adding the third m-sequence and the second m-sequence to generate a secondary scrambling code.

---

[22] See Burns at title and abstract.

As stated above in section 2C, neither Dahlman et al., Dahlman or Burns teach or disclose a scrambling code generator having an adder for adding the third m-sequence and the second m-sequence to generate a secondary scrambling code.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses a scrambling code generator having an adder for adding the third m-sequence and the second m-sequence to generate a secondary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 21 of the present application, of a scrambling code generator having an adder for adding the third m-sequence and the second m-sequence to generate a secondary scrambling code, Claim 21 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 21 under §103(a) must be reversed.

4D. Independent Claim 21 is not rendered obvious by Dahlman et al. in view of Dahlman and Burns

The Examiner has failed to show that each and every element of Claim 21, and in as complete detail as is contained therein, are taught in or suggested by the prior art. The Examiner has failed to make out a prima facia case for an obviousness rejection, and thus Claim 21 is allowable.

5. Dependent Claims 38-47 are also patentable

Dependent Claim 38-47 were said to be unpatentable. Without conceding the patentability per se of dependent Claims 38-47, these claims are likewise believed to be allowable at least by virtue of their dependence on Claim 21.

6. Independent Claim 54 is patentable over Dahlman et al. in view of Dahlman and Burns

Independent Claim 54 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[23]

Claim 54 recites a method for generating scrambling codes in mobile communication system

[23] See Office Action dated March 7, 2007 at page 6, and pages 12-14.

having a scrambling code generator. The method includes generating a $((K-1)*M+K)^{th}$ gold code as a K^{th} primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code. The method further includes generating $((K-1)*M+K+1)^{th}$ through $(K*M+K)^{th}$ gold codes as secondary scrambling codes associated with the K^{th} primary scrambling code. Finally, in the method an L^{th} Gold code is generated by adding an (L-1)-times shifted first m-sequence and a second m-sequence.

Claim 54 relates to method for efficiently dividing the set of Gold sequences into a primary scrambling code set and a secondary scrambling code set to reduce the number of mask functions stored in the memory.

Dahlman et al. discloses slotted mode code usage in a cellular communication system.[24]

Dahlman discloses a method for assigning spreading codes.[25]

Burns discloses a method and apparatus for generating multiple matched-filter pseudo-random noise vectors in a code divisional multiple access demodulator.[26]

6A. Claim 54 recites that the L-th Gold code is generated by adding the (L-1)-times shifted first m sequence and the second m-sequence

Claim 54 recites that the L-th Gold code is generated by adding the (L-1)-times shifted first m sequence and the second m-sequence.

Based on arguments similar to those set forth above in section 2A, neither Dahlman et al., Dahlman or Burns teach or disclose that an L-th Gold code is generated by adding the (L-1)-times shifted first m sequence and the second m-sequence.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 54 of the present application, of generating an L-th Gold code by adding the (L-1)-times shifted first m sequence and the second m-sequence, Claim 54 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 54 under §103(a) must be reversed.

---

[24] See Dahlman et al. at title and abstract.
[25] See Dahlman at title and abstract.

6B. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 54

Claim 54 recites generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code. Thus in Claim 54 a K-th primary scrambling code is generated. A specific gold code is used as the K-th primary scrambling code. The specific gold code is the ((K-1)*M+K)-th gold code. The K-th primary scrambling code is directly related to the total number of secondary scrambling codes per one primary scrambling code.

The Examiner alleges that generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code of Claim 54 is disclosed by Dahlman et al.[27] The Examiner cites Dahlman et al. at col 4, lines 34-57, and Claims 13-23 as disclosing this feature.

Nowhere in Dahlman et al., either the written description or the claims, is there disclosed that a K-th primary scrambling code is directly related to the total number of secondary scrambling codes per one primary scrambling code. Dahlman et al. teaches:[28]

> The scrambling codes typically used in the existing systems are built from, for example, Gold codes, which ensures that the output sequences from the shift register are different for different starting values. Assuming that the normal mode transmission scrambling code, C_j, is generated using a certain starting value, then the scrambling codes to be used for slotted transmissions, $C_{j,1}$ and $C_{j,2}$, can be generated by loading the shift register with a slightly modified value. For example, if two bits in the starting value for the normal mode transmission scrambling code, C_j, are "00", the associated slotted mode scrambling codes, $C_{j,1}$ and $C_{j,2}$, can be generated by changing those two bits in the scrambling code generator shift register to "01" and "11", respectively. As an alternative to loading the shift register with a slightly modified value in order to generate the scrambling codes to be used for slotted transmissions, the same scrambling code as for a normal transmission could be used, but with a different code phase.

---

[26] See Burns at title and abstract.
[27] See Office Action dated March 7, 2007 at bottom of page 12 to the top of page 13.
[28] See Dahlman et al. at col. 4, lines 40-57.

Thus in Dahlman et al., the relation between a primary scrambling code and a secondary scrambling code is a loading of a shift register with a slightly modified value.[29] Dahlman and Burns do not cure these defects.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 54 of the present application, of generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, Claim 54 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 54 under §103(a) must be reversed.

6C. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose generating from ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes as secondary scrambling codes associated with the K-th primary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 54

Claim 54 recites generating from ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes as secondary scrambling codes associated with the K-th primary scrambling code.

Thus in Claim 54 the secondary scrambling codes are generated. Specific gold codes are used as the secondary scrambling codes. The specific gold codes are the ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes. The secondary scrambling codes are directly related to the total number of secondary scrambling codes per one primary scrambling code.

The Examiner again alleges that generating from ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes as secondary scrambling codes associated with the K-th primary scrambling code of Claim 54

[29] See Dahlman et al. at col. 4, lines 45-47.

is disclosed by Dahlman et al.[30] The Examiner uses the same citation for the pervious feature to allegedly reject these features, namely the Examiner cites Dahlman et al. at col 4, lines 34-57, and Claims 13-23 as disclosing this feature.

Nowhere in Dahlman et al., either the written description or the claims, is there disclosed that secondary scrambling codes are directly related to the total number of secondary scrambling codes per one primary scrambling code. Dahlman et al. teaches:[31]

> The scrambling codes typically used in the existing systems are built from, for example, Gold codes, which ensures that the output sequences from the shift register are different for different starting values. Assuming that the normal mode transmission scrambling code, C_j, is generated using a certain starting value, then the scrambling codes to be used for slotted transmissions, $C_{j,1}$ and $C_{j,2}$, can be generated by loading the shift register with a slightly modified value. For example, if two bits in the starting value for the normal mode transmission scrambling code, C_j, are "00", the associated slotted mode scrambling codes, $C_{j,1}$ and $C_{j,2}$, can be generated by changing those two bits in the scrambling code generator shift register to "01" and "11", respectively. As an alternative to loading the shift register with a slightly modified value in order to generate the scrambling codes to be used for slotted transmissions, the same scrambling code as for a normal transmission could be used, but with a different code phase.

Thus again in Dahlman et al., the relation between a primary scrambling code and a secondary scrambling code is a loading of a shift register with a slightly modified value.[32] Dahlman and Burns do not cure these defects.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses generating from ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes as secondary scrambling codes associated with the K-th primary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 54 of the present application, of generating from ((K-1)*M+K+1)-th to (K*M+K)-th Gold codes as secondary scrambling codes associated with the K-th primary scrambling code, Claim 54 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

---

[30] See Office Action dated March 7, 2007 at bottom of page 12 to the top of page 13.
[31] See Dahlman et al. at col. 4, lines 40-57.
[32] See Dahlman et al. at col. 4, lines 45-47.

Based on at least the foregoing, the rejection of independent Claim 54 under §103(a) must be reversed.

6D. Independent Claim 54 is not rendered obvious by Dahlman et al. in view of Dahlman and Burns

The Examiner has failed to show that each and every element of Claim 54, and in as complete detail as is contained therein, are taught in or suggested by the prior art. The Examiner has failed to make out a prima facia case for an obviousness rejection, and thus Claim 54 is allowable.

7. Dependent Claims 55-58 are also patentable

Dependent Claim 55-58 were said to be unpatentable. Without conceding the patentability per se of dependent Claims 55-58, these claims are likewise believed to be allowable at least by virtue of their dependence on Claim 54.

8. Independent Claim 59 is patentable over Dahlman et al. in view of Dahlman and Burns

Independent Claim 59 was said to be unpatentable over Dahlman et al. in view of Dahlman and Burns.[33]

Claim 59 recites an apparatus for generating scrambling codes in mobile communication system having a scrambling code generator. The apparatus includes a first m-sequence generator to generate a first m-sequence. The apparatus further includes a second m-sequence generator to generate a second m-sequence. The apparatus still further includes at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence. Finally, in the apparatus K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Dahlman et al. discloses slotted mode code usage in a cellular communication system.[34]

Dahlman discloses a method for assigning spreading codes.[35]

Burns discloses a method and apparatus for generating multiple matched-filter pseudo-

[33] See Office Action dated March 7, 2007 at page 6 and pages 15-17.
[34] See Dahlman et al. at title and abstract.
[35] See Dahlman at title and abstract.

random noise vectors in a code divisional multiple access demodulator.[36]

8A. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose at least one adder for generating a ((K-1)*M+K)th Gold code as a Kth primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 59

Claim 59 recites at least one adder for generating a ((K-1)*M+K)th Gold code as a Kth primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Thus in Claim 59 a K-th primary scrambling code is generated. A specific gold code is used as the K-th primary scrambling code. The specific gold code is the ((K-1)*M+K)-th gold code. The ((K-1)*M+K)th gold code is generated by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence. The K-th primary scrambling code is directly related to the total number of secondary scrambling codes per one primary scrambling code.

The Examiner alleges that generating a ((K-1)*M+K)-th gold code as a K-th primary scrambling code of Claim 59 is disclosed by Dahlman et al.[37]

As stated above in section 2C, neither Dahlman et al., Dahlman or Burns teach or disclose an adder for adding two m-sequences to generate a scrambling code, namely, at least one adder for generating a ((K-1)*M+K)th Gold code as a Kth primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence.

Also, Claim 59 recites that the time shifting of the first m-sequence is related to ((K-1)*M+K-1). Thus the time shift of the first m-sequence is directly related to the total number of secondary scrambling codes per one primary scrambling code.

Dahlman et al. states that the relation between a primary scrambling code and a secondary

[36] See Burns at title and abstract.
[37] See Office Action dated March 7, 2007 at pages 15-17.

scrambling code is a loading of a shift register with a slightly modified value.[38] Dahlman and Burns do not cure these defects.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof, teaches of discloses at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses at least this recitation of Claim 59 of the present application, of at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, Claim 59 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 59 under §103(a) must be reversed.

8B. Independent Claim 59 is not rendered obvious by Dahlman et al. in view of Dahlman and Burns

The Examiner has failed to show that each and every element of Claim 59, and in as complete detail as is contained therein, are taught in or suggested by the prior art. The Examiner has failed to make out a prima facia case for an obviousness rejection, and thus Claim 59 is allowable.

9. Dependent Claims 60-64 are also patentable

Dependent Claim 60-64 were said to be unpatentable. Without conceding the patentability per se of dependent Claims 60-64, these claims are likewise believed to be allowable at least by virtue of their dependence on Claim 59.

10. Independent Claim 65 is patentable over Dahlman et al. in view of Dahlman and Burns

Independent Claim 65 was said to be unpatentable over Dahlman et al. in view of Dahlman

[38] See Dahlman et al. at col. 4, lines 45-47.

and Burns.[39]

Claim 65 recites a method for generating scrambling codes in mobile communication system having a scrambling code generator. The method includes generating a first m-sequence. The method further includes generating a second m-sequence. The method still further includes generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence. Finally, in the method K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Dahlman et al. discloses slotted mode code usage in a cellular communication system.[40]

Dahlman discloses a method for assigning spreading codes.[41]

Burns discloses a method and apparatus for generating multiple matched-filter pseudo-random noise vectors in a code divisional multiple access demodulator.[42]

10A. Since neither Dahlman et al. nor Dahlman nor Burns teach or disclose generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, neither reference, nor any combination thereof, can be used to render obvious Claim 65

Claim 65 recites generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

As stated above in section 8A, neither Dahlman et al., Dahlman or Burns teach or disclose generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Therefore, neither Dahlman et al., Dahlman nor Burns, or any combination thereof,

---

[39] See Office Action dated March 7, 2007 at page 6 and pages 18-20.
[40] See Dahlman et al. at title and abstract.
[41] See Dahlman at title and abstract.
[42] See Burns at title and abstract.

teaches of discloses generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

Since neither Dahlman et al., Dahlman nor Burns, either alone or in combination, teaches or discloses generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence, wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code, Claim 65 cannot be rendered unpatentable over Dahlman et al., Dahlman nor Burns.

Based on at least the foregoing, the rejection of independent Claim 65 under §103(a) must be reversed.

10B. Independent Claim 65 is not rendered obvious by Dahlman et al. in view of Dahlman and Burns

The Examiner has failed to show that each and every element of Claim 65, and in as complete detail as is contained therein, are taught in or suggested by the prior art. The Examiner has failed to make out a prima facia case for an obviousness rejection, and thus Claim 65 is allowable.

11. Dependent Claims 66-70 are also patentable

Dependent Claim 66-70 were said to be unpatentable. Without conceding the patentability per se of dependent Claims 66-70, these claims are likewise believed to be allowable at least by virtue of their dependence on Claim 65.

CONCLUSION

As the Examiner has failed to make out a prima facie case for an obviousness rejection, the rejection of Claims 1, 21, 31-47 and 54-70 must be reversed.

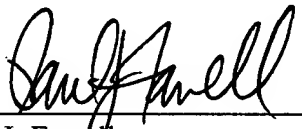
It is well settled that in order for a rejection under 35 U.S.C. §103(a) to be appropriate, the claimed invention must be shown to be obvious in view of the prior art as a whole. A claim may be

found to be obvious if it is first shown that all of the recitations of a claim are taught in the prior art or are suggested by the prior art. In re Royka, 490 F.2d 981, 985, 180 U.S.P.Q. 580, 583 (C.C.P.A. 1974), cited in M.P.E.P. §2143.03.

The Examiner has failed to show that all of the recitations of Claims 1, 21, 31-47 and 54-70 are taught or suggested by the art of record, or the combination thereof. Accordingly, the Examiner has failed to make out a prima facie case for an obviousness rejection.

Independent Claims 1, 21, 54, 59 and 65 are not rendered unpatentable by either Dahlman et al., Dahlman or Burns, or any combination thereof. Therefore, the rejections of Claims 1, 21, 54, 59 and 65 must be reversed.

Dated: September 10, 2007

By: 

Paul J. Farrell
Reg. No.: 33,494
Attorney for Applicant(s)/Appellant(s)

THE FARRELL LAW FIRM, P.C.
333 Earle Ovington Blvd., Suite 701
Uniondale, New York 11553
(516) 228-3565 (tel)
(516) 228-8475 (fax)

CLAIMS APPENDIX

1. (Previously Presented) A method for generating a primary scrambling code, the method comprising the steps of:

generating a first m-sequence from a first m-sequence generator including first shift registers having first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the registers;

generating a second m-sequence from a second m-sequence generator including second shift registers having values b_j, wherein j = 0 to c-1, and where c is the total number of the registers;

masking the first shift register values a_i with a first set of mask values K_i, wherein i = 0 to c-1 to generate a third m-sequence;

adding the first m-sequence with the second m-sequence to generate a primary scrambling code; and

adding the third m-sequence and the second m-sequence to generate a secondary scrambling code;

wherein, the masking step shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code.

2-20. (Cancelled)

21. (Previously Presented) A scrambling code generator, comprising:

a first m-sequence generator to generate a first m-sequence by using a plurality of first registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first registers;

a second m-sequence generator to generate a second m-sequence by using a plurality of second registers with second shift register values b_j, wherein j = 0 to c-1 and where c is the total number of second registers;

a masking section to mask the first shift register values a_i with a first set of mask values K_i to generate a third m-sequence, wherein i = 0 to c-1 to generate a third m-sequence;

a first adder to add the first m-sequence and the second m-sequence to generate a primary scrambling code; and

a second adder to add the third m-sequence and the second m-sequence to generate a secondary scrambling code,

wherein the masking section shifts the first m-sequence cyclically by L chips to generate an L^{th} secondary scrambling code associated with the primary scrambling code.

22-30. (Cancelled)

31. (Previously Presented) The method of claim 1, wherein the primary scrambling code is one of a plurality primary scrambling codes and a K^{th} primary scrambling code is a $((K-1)*M+K)^{th}$ gold code, where M is a total number of secondary scrambling codes per primary scrambling code and $1<K<512$.

32. (Previously Presented) The method of claim 1, wherein the secondary scrambling codes associated with a K^{th} primary scrambling code are from $((K-1)*M+K+1)^{th}$ to $(K*M+K)^{th}$ gold codes, where M is a total number of secondary scrambling codes per primary scrambling code and $1<K<512$.

33. (Previously Presented) The method of claim 1, wherein $1<L<M$, where M is a total number of secondary scrambling codes per primary scrambling code.

34. (Previously Presented) The method of claim 1, wherein the masking step is expressed by $\sum(k_i \times a_i)$.

35. (Previously Presented) The method of claim 1, further comprising:

masking the first shift register values a_i with a second set of mask values K_j to generate a fourth m-sequence, wherein $j = 0$ to $c-1$; and

adding the fourth m-sequence and the second m-sequence to generate an N^{th} secondary

scrambling code associated with the primary scrambling code;

wherein, the masking step shifts the first m-sequence cyclically by N chips to generate an N^{th} secondary scrambling code.

36. (Previously Presented) The method of claim 35, wherein $1<N<M$, where M is a total number of secondary scrambling codes per primary scrambling code.

37. (Previously Presented) The method of claim 1, further comprising the step of delaying at least one of the primary scrambling code and secondary scrambling code to produce a Q-channel component, wherein the primary scrambling code and secondary scrambling code are I-channel components.

38. (Previously Presented) The scrambling code generator of claim 21, wherein the primary scrambling code is one of a plurality of primary scrambling codes and a K^{th} primary scrambling code is a $((K-1)*M+K)^{th}$ gold code, where M is a total number of secondary scrambling codes per primary scrambling code and $1<K<512$.

39. (Previously Presented) The scrambling code generator of claim 38, wherein the secondary scrambling codes associated with the K^{th} primary scrambling code are $((K-1)*M+K+1)^{th}$ to $(K*M+K)^{th}$ gold codes.

40. (Previously Presented) The scrambling code generator of claim 21, further comprising:

a second masking section to mask the first shift register values a_i, with a second set of mask values K_j, wherein $j = 0$ to c-1, to generate a fourth m-sequence; and

a third adder to add the fourth m-sequence and the second m-sequence to generate an N-th secondary scrambling code associated with the primary scrambling code,

wherein the second masking section shifts the first m-sequence cyclically by N chips to generate the N^{th} secondary scrambling code.

41. (Previously Presented) The scrambling code generator of claim 21, wherein the masking section shifts the first m-sequence cyclically by masking the first shift register values a_i in accordance with $\sum(K_i \times a_i)$.

42. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator cyclically shifts the first shift register values and the second m-sequence generator cyclically shifts the second shift register values.

43. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator adds predetermined shift register values of the first shift registers based on a first generating polynomial of the first m-sequence, right shifts the first shift register values a_i of the first shift registers, and replaces the first register value a_{c-1} with the result of the addition of the predetermined register values.

44. (Previously Presented) The scrambling code generator of claim 21, wherein the first m-sequence generator adds a first shift register value a_0 with a first shift register a_7 to form a next first shift register a_{c-1}.

45. (Previously Presented) The scrambling code generator of claim 21, wherein the second m-sequence generator adds predetermined shift register values of the second shift registers based on a second generating polynomial of the second m-sequence, right shifts the second shift register values b_j of the second shift registers, and replaces the second register value b_{c-1} with the result of the addition of the predetermined register values.

46. (Previously Presented) The scrambling code generator of claim 21, wherein the second m-sequence generator adds a second shift register value b_0 with a second shift register value b_5, b_7, and a second shift register value b_{10} to form a next second shift register value b_{c-1}.

47. (Previously Presented) The apparatus of claim 21, further comprising a means for

delaying at least one of the primary scrambling code and the secondary scrambling code to produce Q-channel component, wherein the primary scrambling code and the secondary scrambling code are I-channel components.

48 - 53. (Cancelled)

54. (Previously Presented) A method for generating scrambling codes in mobile communication system having a scrambling code generator, the method comprising steps of:

generating a $((K-1)*M+K)^{th}$ gold code as a K^{th} primary scrambling code, where K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code; and

generating $((K-1)*M+K+1)^{th}$ through $(K*M+K)^{th}$ gold codes as secondary scrambling codes associated with the K^{th} primary scrambling code,

wherein an L^{th} Gold code is generated by adding an (L-1)-times shifted first m-sequence and a second m-sequence.

55. (Previously Presented) The method as claimed in claim 54, wherein K is a primary scrambling code number and $1 \leq K \leq 512$.

56. (Previously Presented) The method as claimed in claim 55, wherein the first m-sequence is generated from a first shift register memory having a plurality of first shift registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first registers and the (L-1)-times shifted first m-sequence is generated by masking the first shift register values a_i with mask values K_i, where i = 0 to c-1.

57. (Previously Presented) The method as claimed in claim 56, wherein the masking is performed according to: $\sum(K_i \times a_i)$.

58. (Previously Presented) The method as claimed in claim 54, wherein the generated primary scrambling code and secondary scrambling code are I-channel components and the method further comprises delaying at least one of the primary scrambling code and secondary scrambling code to produce Q-channel components.

59. (Previously Presented) An apparatus for generating scrambling codes in mobile communication system having a scrambling code generator, comprising:

a first m-sequence generator to generate a first m-sequence;

a second m-sequence generator to generate a second m-sequence; and

at least one adder for generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence,

wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

60. (Previously Presented) The apparatus of claim 59, wherein the secondary scrambling codes of the K^{th} primary scrambling codes are the $((K-1)*M+K+1)^{th}$ through $(K*M+K)^{th}$ Gold codes.

61. (Previously Presented) The apparatus as claimed in claim 60, wherein K is a primary scrambling code number and $1 \leq K \leq 512$.

62. (Previously Presented) The apparatus as claimed in claim 59, wherein the first m-sequence generator comprises a plurality of first registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first shift registers, and the scrambling code generator further comprising at least one masking section for generating the n-times shifted first m-sequence by masking the first shift register values a_i with mask values K_i, where i = 0 to c-1.

63. (Previously Presented) The apparatus as claimed in claim 62, wherein the masking is performed according to: $\sum(K_i \times a_i)$.

64. (Previously Presented) The apparatus as claimed in claim 59, wherein the primary scrambling code and secondary scrambling code are I-channel components and the apparatus further comprises a means for delaying at least one of the primary scrambling codes and secondary scrambling code to produce Q-channel components.

65. (Previously Presented) A method for generating scrambling codes in mobile communication system having a scrambling code generator, comprising the steps of:

generating a first m-sequence;

generating a second m-sequence; and

generating a $((K-1)*M+K)^{th}$ Gold code as a K^{th} primary scrambling code by adding a ((K-1)*M+K-1)-times shifted first m-sequence and the second m-sequence,

wherein K is a natural number and M is a total number of secondary scrambling codes per one primary scrambling code.

66. (Previously Presented) The method as claimed in claim 65, further comprising generating $((K-1)*M+K+1)^{th}$ to $(K*M+K)^{th}$ Gold codes as secondary scrambling codes corresponding to the K^{th} primary scrambling code.

67. (Previously Presented) The method as claimed in claim 65, wherein K is a primary scrambling code number and $1 \leq K \leq 512$.

68. (Previously Presented) The method as claimed in claim 65, wherein the first m-sequence is generated from a first shift register memory having a plurality of first shift registers with first shift register values a_i, wherein i = 0 to c-1 and where c is the total number of the first registers and the n-times shifted first m-sequence is generated by masking the first shift register values a_i with mask values K_i, where i = 0 to c-1.

69. (Previously Presented) The method as claimed in claim 68, wherein the masking is performed according to: $\sum(K_i \times a_i)$.

70. (Previously Presented) The method as claimed in claim 65, wherein each scrambling code is used as an I-channel component and a Q-channel component, corresponding to the I-channel component, is generated by delaying the I-channel component for a predetermined time.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to 37 C.F.R. 1.130, 1.131, 1.132 or entered by the Examiner and relied upon by Appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 C.F.R. 41.37.